

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 28 August 2022

C. Bormann
Universität Bremen TZI
24 February 2022

Using CDDL for CSVs
draft-bormann-cbor-cddl-csv-00

Abstract

The Concise Data Definition Language (CDDL), standardized in [RFC 8610](#), is defined to provide data models for data shaped like JSON or CBOR.

Another representation format that is quite popular is the CSV file as defined by [RFC 4180](#).

The present document shows how to use CDDL to provide a data model for CSV files.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 28 August 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

Internet-Draft

CDDL for CSVs

February 2022

extracted from this document must include Revised BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Revised BSD License.

Table of Contents

1.	Introduction	2
1.1.	Terminology	2
2.	CSV generic data model	2
3.	Examples	3
4.	IANA Considerations	4
5.	Security considerations	5
6.	References	5
6.1.	Normative References	5
6.2.	Informative References	5
	Acknowledgements	5
	Author's Address	6

[1.](#) Introduction

The Concise Data Definition Language (CDDL), standardized in [\[RFC4180\]](#), is defined to provide data models for data shaped like JSON or CBOR.

Another representation format that is quite popular is the CSV file as defined by [\[RFC4180\]](#).

The present document shows how to use CDDL to provide a data model for CSV files.

[1.1.](#) Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#) [\[RFC2119\]](#) [\[RFC8174\]](#) when, and only when, they appear in all capitals, as shown here.

This specification uses terminology from [\[RFC8610\]](#).

[2.](#) CSV generic data model

The CSV format is defined in [\[RFC4180\]](#). The generic data model for

the data in a CSV file can be described in CDDL as:

```
csv = [?header, *record]
header = [+header-field]
record = [+field]
header-field = text
field = text
```

Note that the elements of this data model describe the interpretation of the data after removal of lexical structure such as newlines, commas, escape characters, and quotation marks.

For the purposes of a specific application, the data model level structure of each field may be described in a more elaborate way, e.g., as a number. CDDL currently does not have a way to express the transformation from the text string in the CSV field to the number that this text string represents at the application data model level; the usage of anything but "text" for a field therefore **MUST** be accompanied by an instruction how to perform the translation. As a preferred choice, the JSON representation of the data model item, if it exists, **MAY** be chosen by that instruction.

Since the CSV media type `text/csv` defaults to `us-ascii` (see [Section 3 of \[RFC4180\]](#)), many uses of CSV will need to specify the media type parameter `charset`. The media type parameter `header` **MAY** be used to indicate the presence or absence of a header line; if it is not given, the grammar **MUST NOT** be ambiguous about the presence of a header (i.e., it **MUST** be either mandatory or absent).

Note that the ABNF [\[STD68\]](#) in [\[RFC4180\]](#) does not quite handle the case that `charset` is not `us-ascii`. For the purposes of the present specification, the ABNF is understood to allow all characters from the `charset` except `%x22` and `%x2C` in `TEXTDATA`. For the purposes of the present specification, the ABNF rule `CRLF` is read as:

```
CRLF = [CR] LF
```

as is hinted in [Section 3 of \[RFC4180\]](#).

3. Examples

A simplified CSV form definition of a SID file [[I-D.ietf-core-sid](#)] might look like this:

```
; header = absent
```

```
SID-File = [meta-record,  
            *dependency-record,  
            *range-record,  
            *item-record]
```

```
meta-record = ["meta",  
               module-name: text,  
               module-revision: empty / text,  
               sid-file-revision: empty / text,  
               description: empty / text]
```

```
dependency-record = ["dep",  
                     module-name: text,  
                     module-revision: text]
```

```
range-record = ["range",  
                entry-point: uint,  
                size: uint]
```

```
item-record = ["item",  
               namespace: "module" / "identity" / "feature" / "data",  
               identifier: yang-identifier / schema-node-path  
               ; the above probably should say which namespace  
               ; goes with which identifier  
               sid: uint]
```

```
yang-identifier = text .abnf ("yang-identifier" .det id-abnf)
```

```

schema-node-path = text .abnf ("schema-node-path" .det id-abnf)
id-abnf = '
    schema-node-path = QID *( "/" OQID)
    yang-identifier = ID
    QID = ID ":" ID
    OQID = ID [ ":" ID]
    ID = I *C
    I = "_" / %x41-5a / %x61-7a
    C = I / %x30-39 / "-" / "."
    ,

```

empty = ""

TODO: show the example in [Appendix A](#) of [[I-D.ietf-core-sid](#)]

[4.](#) IANA Considerations

This document makes no requests of IANA.

[5.](#) Security considerations

The security considerations of [[RFC8610](#)] and [[RFC4180](#)] apply.

[6.](#) References

[6.1.](#) Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4180] Shafranovich, Y., "Common Format and MIME Type for Comma-Separated Values (CSV) Files", [RFC 4180](#), DOI 10.17487/RFC4180, October 2005, <<https://www.rfc-editor.org/info/rfc4180>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", [RFC 8610](https://www.rfc-editor.org/info/rfc8610), DOI 10.17487/RFC8610, June 2019, <<https://www.rfc-editor.org/info/rfc8610>>.
- [STD68] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](https://www.rfc-editor.org/info/rfc5234), DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.

[6.2.](#) Informative References

- [I-D.ietf-core-sid] Veillette, M., Pelov, A., Petrov, I., Bormann, C., and M. Richardson, "YANG Schema Item iDentifier (YANG SID)", Work in Progress, Internet-Draft, [draft-ietf-core-sid-18](https://www.ietf.org/archive/id/draft-ietf-core-sid-18), 18 November 2021, <<https://www.ietf.org/archive/id/draft-ietf-core-sid-18.txt>>.

Acknowledgements

Rob Wilton, unknowingly, made me write this specification. I hope it will be useful.

Author's Address

Carsten Bormann
Universität Bremen TZI
Postfach 330440
D-28359 Bremen
Germany
Phone: +49-421-218-63921
Email: cabo@tzi.org

